Kitchen Timer: Phase B

ENCE 4231

Madeline Bohn

GitHub Link: https://github.com/madeline-bohn/ENCE_3220_Class2023

Project Requirements

This project focuses on constructing a kitchen timer that counts down from a user specified value. The final protect needs to have a readable display so that the time left can be viewed by the user. Additionally, it needs a sound mechanism to alert that the time is up. The timer also needs a user interface so that the time being counted down from can be dictated.

System Design

The design of the system is laid out below in the block diagram (Figure 1). The clock system is computationally run via a microprocessor which outputs to user LEDs to indicate its current state, a 7-segment display to communicate the time left, and a buzzer to alert the user that the timer is finished. The microprocessor has several inputs including a crystal oscillator, a USB so that power can be received from a laptop device, a Wi-Fi (comms) module so that alternatively to the buttons the timer can be controlled via a website, a voltage regulator, and lastly, an ISP programmer. These components are to be soldered to a PCB board and enclosed in a 3D printed enclosure.

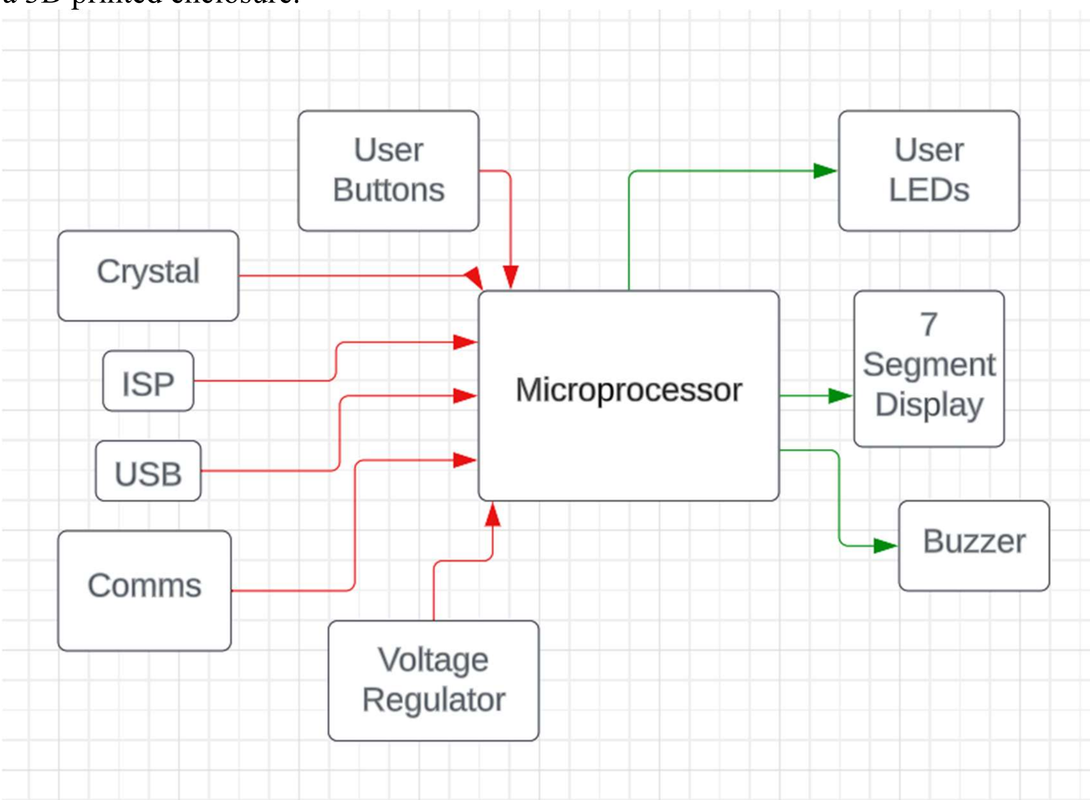

Figure 1: Kitchen Timer Block Diagram

Components Selection

The full itemized list of components can be found in Figure 2. The components were selected based on their ability to complete the necessary functions described in the project requirements and their compatibility with one another based on the block diagram above while maintaining a reasonable price. The most important component selection made was the microprocessor. The ATmega32U4-A was chosen as it is smaller in size but with the same, necessary computational abilities of an Arduino. Additionally, the display included a green and red LED to indicate the start, stop, and incrementation of the timer, a buzzer (Piezo Buzzers & Audio Indicators buzzer, 12 mm x 9.5 mm at 85dB) that is loud enough to notify that the timer is done but not excessive, and lastly, a 7-segment display that can show up to 4 digits since this timer is only minutes and seconds. A programmable ESP WiFi module was chosen as a way to interface with a website to control the timer. Additionally, two push buttons were chosen as a means of controlling the timer physically. A variety of other components were necessary in order to execute the necessary functions such as shift registers, voltage regulator, resistors etc. The full list can be seen below.

	Mouser #	Mfr. #	Manufacturer	Customer #	Description
1	511-USBLC6-2SC6	USBLC6-2SC6	STMicroelectronics		ESD Suppressors / TVS Diodes ESD Protection Low Cap
2	556-ATMEGA32U4-AU	ATMEGA32U4-AU	Microchip		8-bit Microcontrollers - MCU AVR USB 32K FLASH INDUSTRIAL
3	595-LP2985-33DBVR	LP2985-33DBVR	Texas Instruments		LDO Voltage Regulators 150-mA Low-Noise 1.5% tolerance
4	652-MF-MSMF050-2	MF-MSMF050-2	Bourns		Resettable Fuses - PPTC 15V .5A-HD 100A MAX
5	595-SN74HC595PWRG4	SN74HC595PWRG4	Texas Instruments		Counter Shift Registers 8B Shift Registers
6	815-ABM816000MHZD1XT	ABM8-16.000MHZ-D1X-T	ABRACON		Crystals CRYSTAL 16.0000MHZ 18PF SMD
7	611-PTS526SK08SMTR2L	PTS526 SK08 SMTR2 LFS	C&K Switches		Tactile Switches 50mA 12VDC, 5.2x5.2mm, 0.8mm H, 260gf, G leads, No ground pin, no actuator
8	611-PTS125M73SMTRLFS	PTS125SM73SMTR21M LFS	C&K Switches		Tactile Switches SWITCH TACTILE
9	490-UJ2-MBH-4-SMT	UJ2-MBH-4-SMT-TR	CUI Devices		USB Connectors USB 2.0 mini B jack 5 pin Horizontal SMT
10	581-06035C104KAT2A	06035C104KAT2A	KYOCERA AVX		Multilayer Ceramic Capacitors MLCC - SMD/SMT 50V 0.1uF X7R 0603 10%
11	581-06031A220FAT2A	06031A220FAT2A	KYOCERA AVX		Multilayer Ceramic Capacitors MLCC - SMD/SMT 100V 22pF C0G 0603 1%
12	581-06035C105KAT2A	06035C105KAT2A	KYOCERA AVX		Multilayer Ceramic Capacitors MLCC - SMD/SMT 50V 1uF X7R 0603 10%
13	581-0805YC106KAT2A	0805YC106KAT2A	KYOCERA AVX		Multilayer Ceramic Capacitors MLCC - SMD/SMT 16V 10uF X7R 0805 10%
14	581-06036D225K	06036D225KAT2A	KYOCERA AVX		Multilayer Ceramic Capacitors MLCC - SMD/SMT 6.3V 2.2uF X5R 0603 10%
15	581-06033C103KAT2A	06033C103KAT2A	KYOCERA AVX		Multilayer Ceramic Capacitors MLCC - SMD/SMT 25V 0.01uF X7R 0603 10%
16	756-WCR0805-100RFA	WCR0805-100RFA	TT Electronics		Thick Film Resistors - SMD 0805 100 Ohm 1% Tol AEC-Q200
17	279-CPF0805B10KE1	CPF0805B10KE1	TE Connectivity		Thin Film Resistors - SMD CPF 0805 10K 0.1% 25PPM
18	71-RCC0805330RFKEA	RCC0805330RFKEA	Vishay		Thick Film Resistors - SMD 1/4W 330ohms 1% 100ppm
19	603-RT0603FRE0710KL	RT0603FRE0710KL	YAGEO		Thin Film Resistors - SMD 10K ohm 1% 1/10W
20	652-CHP0603QJW-220E	CHP0603QJW-220ELF	Bourns		Thick Film Resistors - SMD ResHigh PowerQ 0603 22R 5% 1/3W TC200
21	652-CR0603FX-1001ELF	CR0603-FX-1001ELF	Bourns		Thick Film Resistors - SMD 1K 1% 1/10W
22	710-150080BS75000	150080BS75000	Wurth Elektronik		Standard LEDs - SMD WL-SMCW SMDMono TpVw Waterclr 0805 Blue
23	710-150080RS75000	150080RS75000	Wurth Elektronik		Standard LEDs - SMD WL-SMCW SMDMono TpVw Waterclr 0805 Red
24	710-150080VS75000	150080VS75000	Wurth Elektronik		Standard LEDs - SMD WL-SMCW SMDMono TpVw Waterclr 0805 BrtGrn
25	490-CMI-1295-0585T	CMI-1295-0585T	CUI Devices		Piezo Buzzers & Audio Indicators buzzer, 12 mm x 9.5 mm deep, M, 5 V, 85 dB, Through Hole, Audio Indicator
26	859-LTC-5723HR	LTC-5723HR	Lite-On		LED Displays & Accessories 4 Digit, Red
27	649-10056844-106ALF	10056844-106ALF	Amphenol		Headers & Wire Housings QKE HDR SS STR DP
28	649-68001-416HLF	68001-416HLF	Amphenol		Headers & Wire Housings 16P SR UNSHRD HRD TIN OVER NI

Figure 2: Component Order List (provided by instructor)

Build Prototype

The prototype was constructed in Phase A using an Arduino Uno and Arduino Uno shield with PCB's printed from instructor designs. This allowed for continuity between those working with the prototypes as well as for code development to begin without the Phase B prototypes printed. The code for the prototype included most the needed logic and just had to be merged with the ESP WiFi module code in order to complete Phase B. Using an Arduino to prototype is helpful as there exists a plethora of resources on how to program them. Prototyping this way allowed for the software development in Phase B to happen more seamlessly.

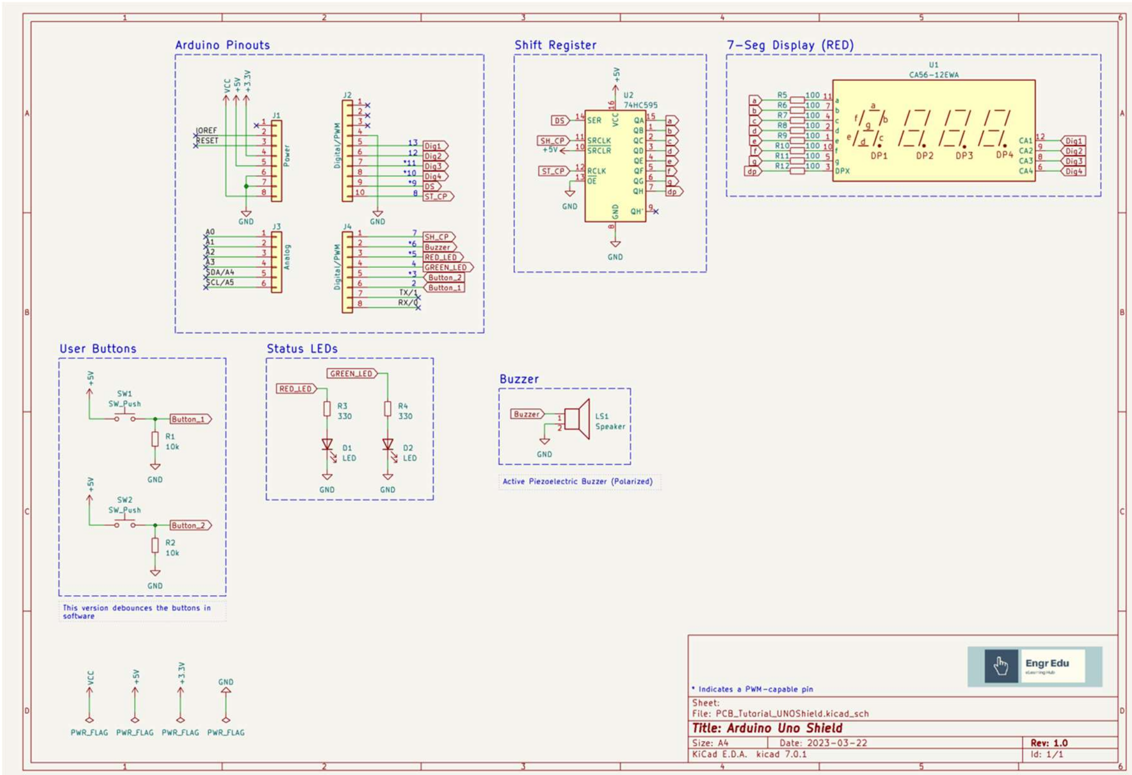

Figure 3: Prototype Schematic from Phase A (provided by instructor)

PCB Design

The next step of the project consisted of the PCB design which was completed in the software KiCad. The schematic in Figure 3 was generated using this software and shows the implementation of the block diagram and the components selected. Once the schematic was completed with the proper components, wiring, and footprints, a PCB was generated and designed from it (Figure 4). This 2D layout can then be used to generate a 3D render of the board so that the appropriate modifications can be made to both the board and the enclosure designs (Figures 5 and 6). From the PCB design, the appropriate files can be generated so that the board can be fabricated and soldered.

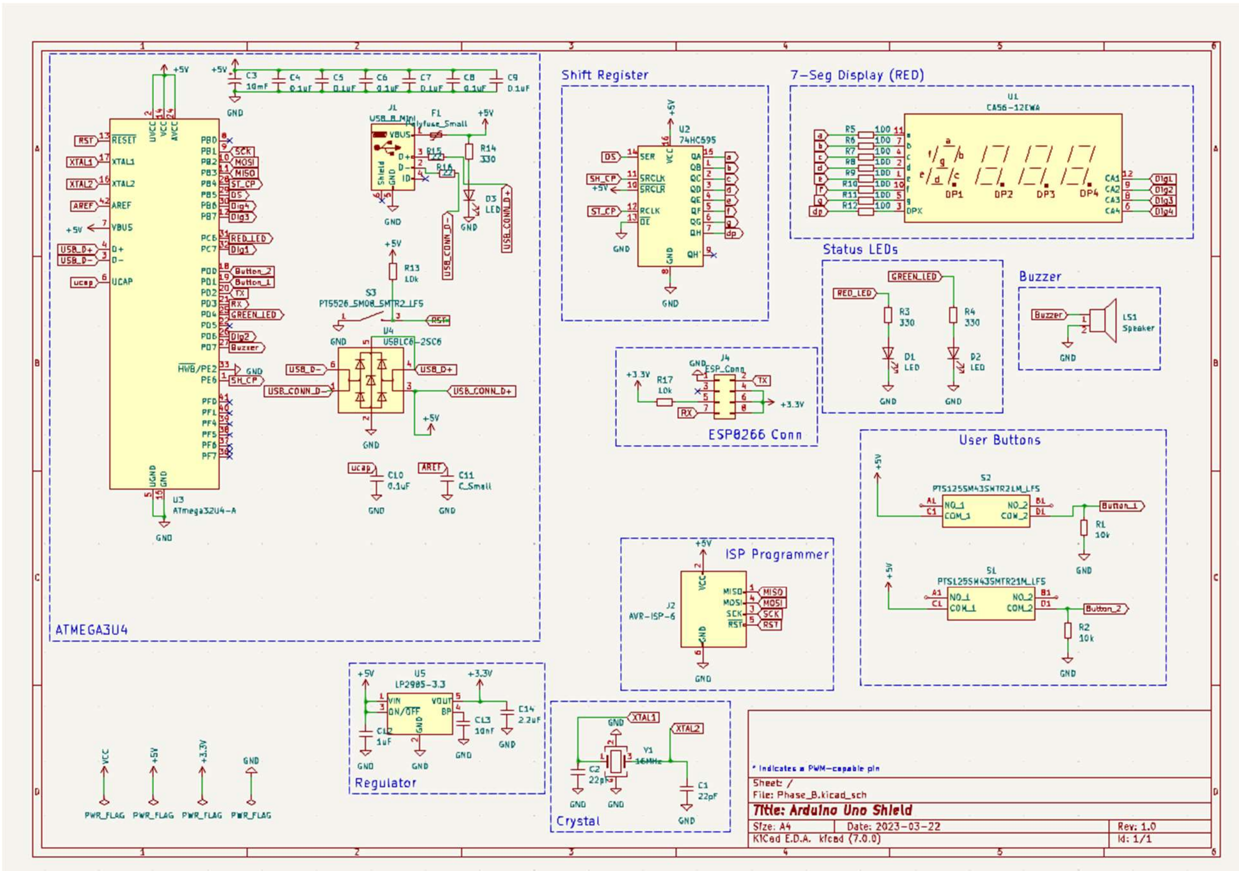

Figure 4: Schematic Designed in KiCad

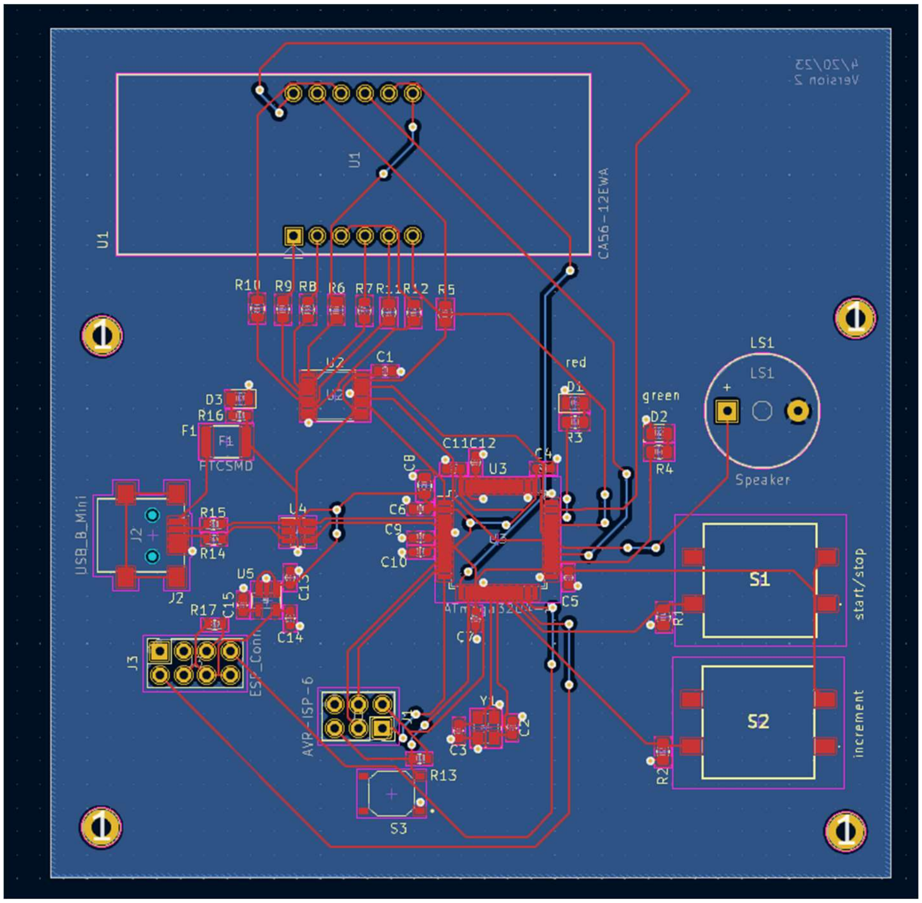

Figure 5: PCB Layout Design

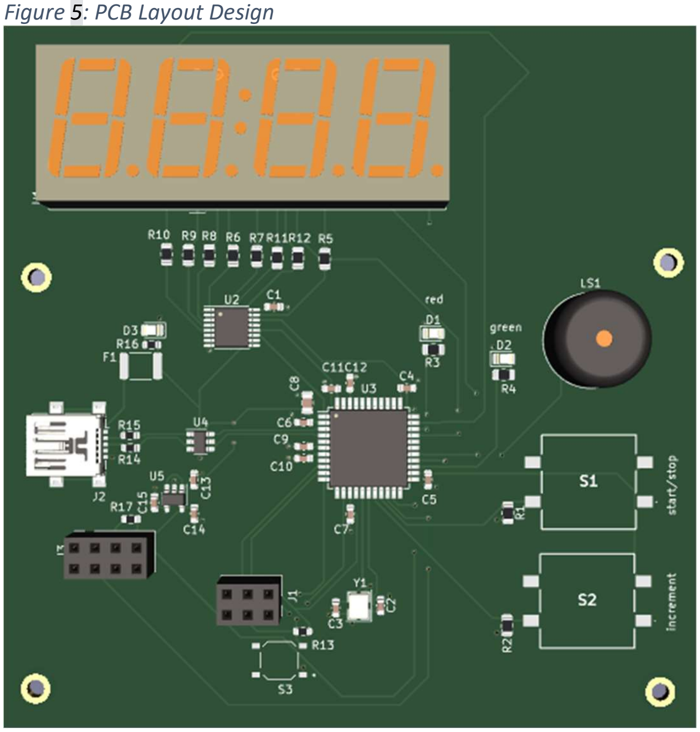

Figure 6: 3D Render of Front of PCB

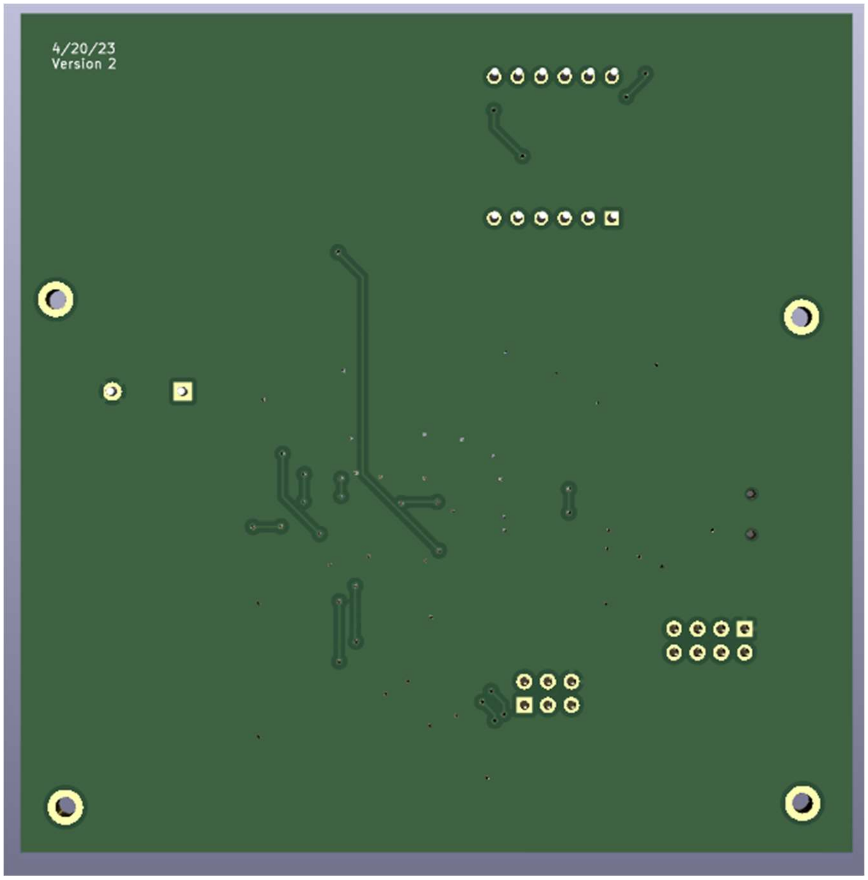

Figure 7: 3D Render of the Back of the PCB

Assemble Stage

To be completed.

Software Development

The code for this system required the integration of the ESP WiFi module into the Phase A prototype code. Together, the code contains a variety of helper functions that are called in the main loop in conjunctions with interrupts and their handlers for components such as buttons and the ESP module. The flow chart below describes in detail the logic flow of the code. The actual Arduino code has been uploaded to Github. The most difficult part of this process was merging the loops of the two scripts as determining the order of all the nested if/else statements was difficult. Additionally, understanding the timer syntax and setup (Figure 7) was difficult.

```
// Initialize Timer1
// Speed of Timer1 = 16MHz/64 = 250 KHz
TCCR1A = 0;
TCCR1B = 0;
OCR1A = gReloadTimer1;            // max value 2^16 - 1 = 65535
TCCR1A |= (1<<WGM11);
TCCR1B = (1<<CS11) | (1<<CS10);   // 64 prescaler
TIMSK1 |= (1<<OCIE1A);
interrupts();
```

Figure 8: Timer Initialization Code

```
void loop() {
  char messageBuff[BUFF_SIZE];
  int auxCount = 0;
  unsigned char auxDigit = '0';


  if(gISRFlag1 == 1 || gISRFlag2 == 1){
    // Reset ISR Flag
    gISRFlag1 = 0;
    gISRFlag2 = 0;

    gIncomingChar = Serial.read(); //read incoming data

    if(gPackageFlag == 1){
      gCommMsg[buff] = gIncomingChar;  //if new message add to buffer and increment size
      buff++;

      if(buff == BUFF_SIZE){
        gPackageFlag = 0; //if at max size reset start flag and raise processor flag
        gProcessFlag = 1;
      }
    }
    if(gIncomingChar == '$'){
      gPackageFlag = 1; //if start of msg -> raise flag

      int i;
      for(int i = 0; i<BUFF_SIZE; i++){
        gCommMsg[i] == 0; //set message to 0
      }
      buff = 0; //set msg index to 0
    }

    if((gIncomingChar == '\n') && (gPackageFlag == 1)){
      gPackageFlag = 0; //signal end of msg
      gProcessFlag == 1;
    }
    if(gTimerRunning == 0){
      // Start Timer
      gTimerRunning = 1;

      if(gCount == 0){
        gCount = DEFAULT_COUNT;
      }
      if(gBuzzerFlag == 1){
        gBuzzerFlag = 0;

        // LEDs -> Timer Stopped
        digitalWrite(RED_LED, HIGH);
        digitalWrite(GREEN_LED, HIGH);
      }
      else
      {
        startTimer1();
        // LEDs -> Timer Running
        digitalWrite(RED_LED, LOW);
        digitalWrite(GREEN_LED, HIGH);
      }
    }
```

Figure 9: Excerpt from loop()

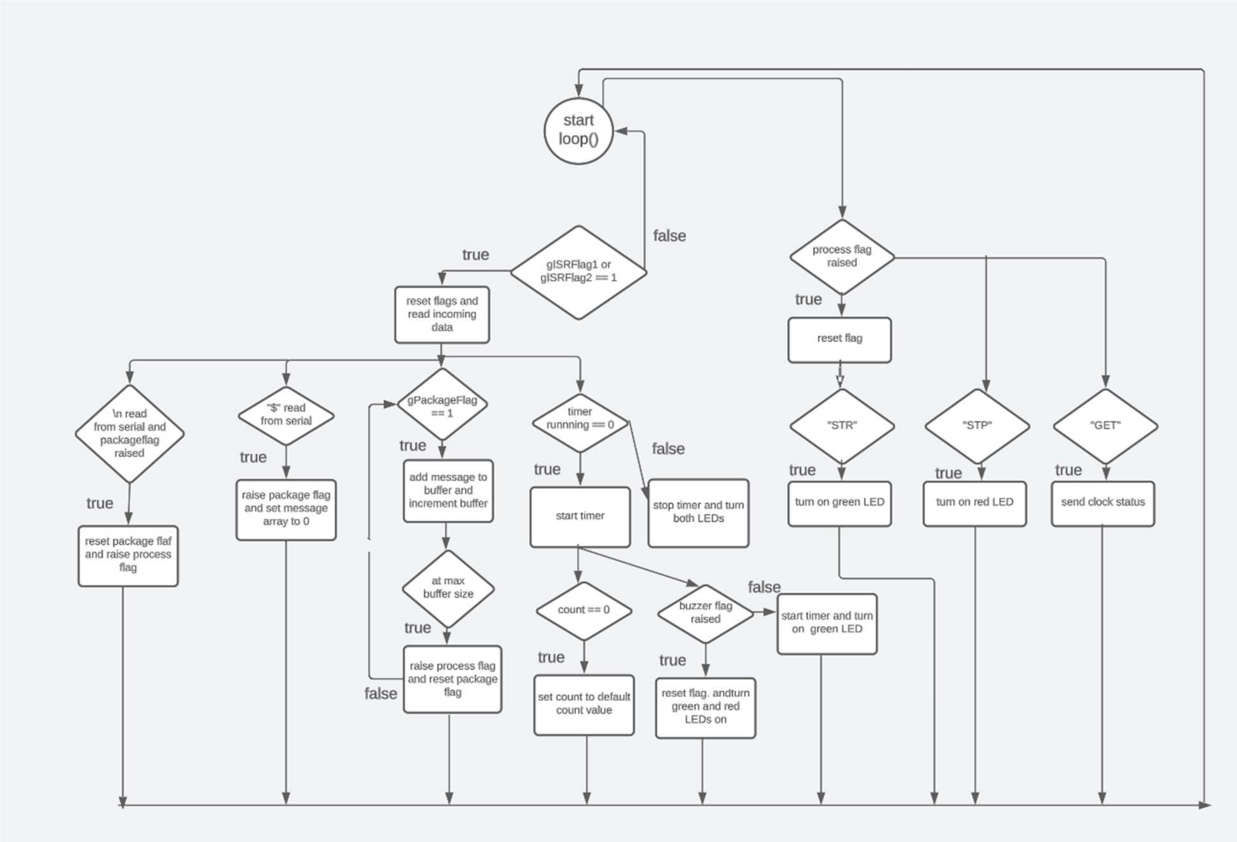

Figure 10: Flow Chart for Kitchen Timer Code

Enclosure Design

For aesthetic and functional purposes, it is desirable to create an enclosure for the PCB. One method is to design the enclosure using the PCB layout in a CAD software such as SolidWorks to generate a model that can be 3D printed. Figure 8 shows a sample enclosure design based on the PCB design in Figure 5. Figure 7 shows what this PCB design looks like inside the enclosure and how the 3D renders fit together. The enclosure provides a view of the 7-segment display as well as the LEDs and access to the buttons and USB port.

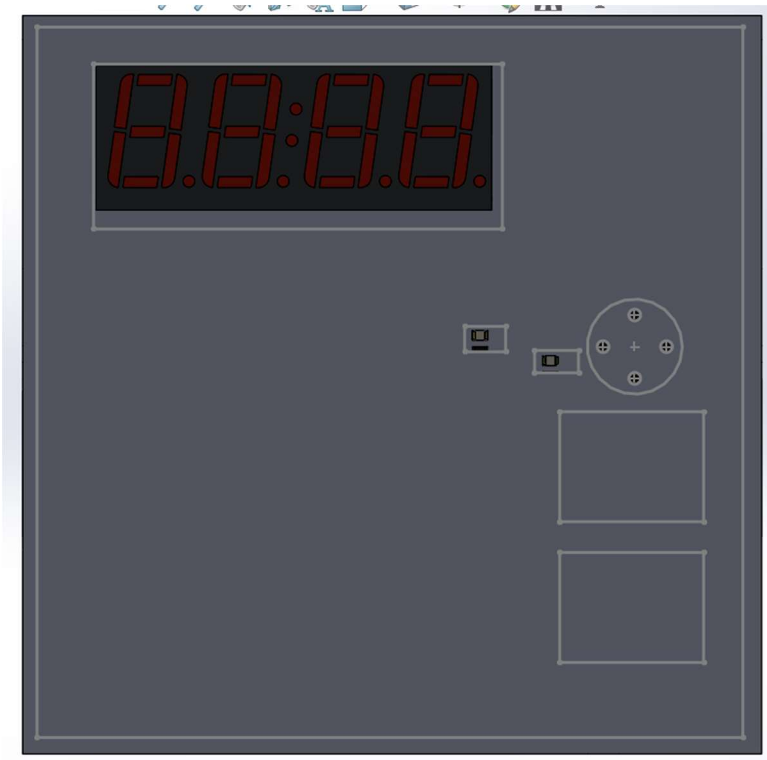

Figure 11: Enclosure Assembly Design with SolidWorks

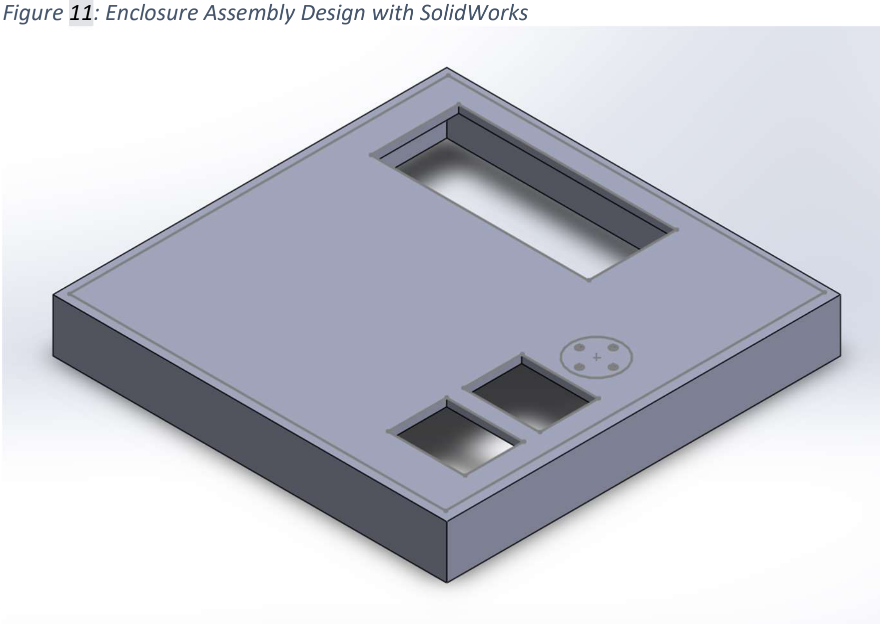

Figure 12: Enclosure Design Generated with SolidWorks